

廃棄厳禁 (ファイルに上綴じ状態で保存)

# 国内中間処理に伴うIDS提出要否調査結果シート (資料33)

改版: 2003. 6. 3 / 黒川

国内中間処理に伴う対応US出願有無調査結果において、「対応US出願・有」かつ「IDS提出要否調査依頼・要」にマークが付されている場合は、未提出引用文献などのIDS提出が指令日(発送日)から3ヶ月以内に必要になります。この場合、技術担当者は直ちにIDS提出担当者(現: 大川さん)に指令書とともにこのシートを送付し、IDS提出要否調査を行ってもらって下さい。詳細は資料34を参照。

対応US出願有無調査	P- 10/483	技術担当者:
調査依頼日:	依頼者(分配管理責任者):	
回答日: 2010年4月7日	調査者(現: 山部 さん): 山部	
調査方法: (桐) データベース / 顧客確認要 / (その他) 特許情報 記載あり		
対応US出願有無: (有) / 無(有の場合、OP番号、その状況、技術担当者を記載) OP-C4089-PC-US → 係属中。技術担当者: 中村 (10/581054)		
IDS提出要否調査依頼: 要 / 否 / 保留(技担が顧客確認後に要否決定)		

IDS提出要否調査	
調査依頼日:	依頼者(技術担当者):
回答日: 2010年4月7日	調査者(現: 大川さん): 山部 山部
IDS提出要否: (要) / 否(要の場合、技術担当者は直ちにIDS提出処理依頼をIDS提出担当者に行う。)	

技術担当者の最終指示	英文抄録提出 / その他 ( )
------------	------------------

JP08-212064, JP05-307478は未提出

JP02-139630は 2009年6月25日付IDS提出済

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-212064

(43)Date of publication of application : 20.08.1996

(51)Int.Cl. G06F 9/06  
G06F 9/46

(21)Application number : 07-014524 (71)Applicant : HITACHI LTD

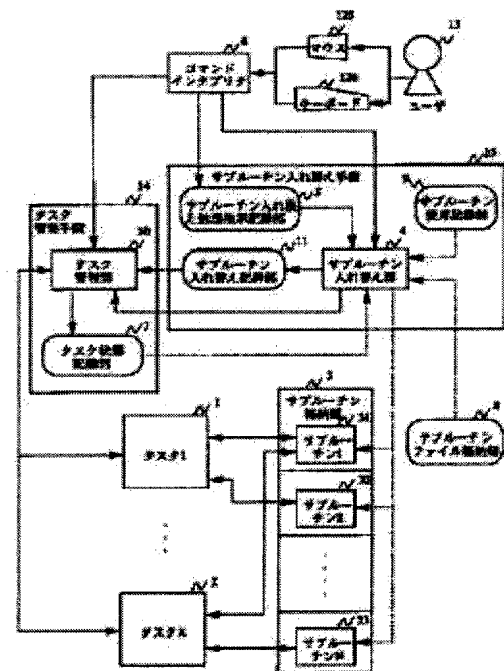
(22)Date of filing : 31.01.1995 (72)Inventor : TSUNETOMI  
KUNIIHIKO  
YOKOYAMA  
TAKANORI  
SHIMADA MASARU  
SAITO MASAHIKO  
NAKAMURA  
TOMOAKI

## (54) SWITCHING SYSTEM AND METHOD FOR SHARED SUBROUTINE

### (57)Abstract:

**PURPOSE:** To effectively switch the shared subroutines without stopping the operation of an entire system.

**CONSTITUTION:** A shared subroutine switching system includes a subroutine switch processing instruction recorder means 5 which instructs how to deal with the task that is using a subroutine when a subroutine switching command is carried out. Then the system decides the using state of the subroutine to be switched and carries out the switch processing when the subroutine is not used by any task and also the means 5 has no designation. In other cases, the task is compulsively finished by an instruction or the switch processing is delayed until the task is finished and the switch processing becomes possible.





## 【特許請求の範囲】

【請求項1】一つ以上のタスクと、タスク間で共有して実行可能な共用サブルーチンと、該共用サブルーチンを格納するサブルーチン格納部と、当該サブルーチン格納部に格納された共用サブルーチンを入れ替えるサブルーチン入れ替え手段と、タスクの実行開始処理と実行終了処理を行なうタスク管理手段を有する、共用サブルーチン入れ替えシステムにおいて、

該サブルーチン入れ替え手段は、サブルーチンの入れ替えを行う際に、入れ替えの対象となるサブルーチンがタスクにより使用中であった場合の該サブルーチンを使用するタスクに対する処理を指定するサブルーチン入れ替え処理指示記録手段を有し、

該タスク管理手段は、タスクが使用するサブルーチンの入れ替え処理が実行中か判定し、入れ替え処理中ならば該サブルーチンを使用するタスクの処理を中断乃至は強制終了し、

上記サブルーチン入れ替え手段は、入れ替えの対象となる共用サブルーチンの使用状況を判定し、当該共用サブルーチンがタスクに使用されておらず、かつサブルーチン入れ替え処理記録手段に指定がない場合、入れ替え処理を行ない、それ以外の場合には、該サブルーチン入れ替え処理指示記録手段に記録された指示に従って処理を行うことを特徴とする、共用サブルーチン入れ替えシステム。

【請求項2】一つ以上のタスクと、タスク間で共有して実行可能な共用サブルーチンと、該共用サブルーチンを格納するサブルーチン格納部とを有する計算機システムにおける共用サブルーチン入れ替え方法であって、共用サブルーチンを入れ替える際に、入れ替えの対象となるサブルーチンがタスクにより使用中であるか否かを判定する判定処理と、

該判定処理により、該サブルーチンがタスクにより使用中である場合は、該サブルーチンを使用するタスクの性質により、該タスクに対する操作を決定する処理を有することを特徴とする共用サブルーチン入れ替え方法。

【請求項3】請求項2の共用サブルーチン入れ替え方法における、前記サブルーチンを使用するタスクの性質は、タスクが処理の終了後に自ら停止するか否かを元に判定し、該タスクが処理の終了後に自ら停止する場合には、前記タスクに対する操作を決定する処理において該タスクが停止するまでサブルーチンの入れ替えを中断することを特徴とする共用サブルーチン入れ替え方法。

【請求項4】請求項3の共用サブルーチン入れ替え方法において、上記タスクの性質はタスクのコンパイル時にタスクのプログラムを元に判定しておくことを特徴とする共用サブルーチン入れ替え方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、複数のタスクが複数の

共用サブルーチンを共有しているシステムにおいて、システム全体を停止させることなく共用サブルーチンの入れ替えを行う、共用サブルーチン入れ替えシステムおよび共用サブルーチンの入れ替え方法に関する。

## 【0002】

【従来の技術】ユニックス (UNIX: ユニックス システム ラボラトリーズ (Unix System Laboratories) のライセンスしているオペレーティングシステム) 等の従来のオペレーティングシステムは、システム内の全てのタスク間で共用して使用することが可能な共用サブルーチン (ユニックスでは共用ライブラリと呼ばれる) を用意している。共用サブルーチンはタスクのオブジェクトファイルと一緒に含まれずに別ファイルとなっている。この為、全てのタスクのオブジェクトファイルにサブルーチンを予めリンクしておく場合に比べて、オブジェクトファイルを格納するハードディスクの使用量を削減できる。また、メモリ上の共用サブルーチンも複数のタスクにより共用されるので、メモリ使用量も節約できる。

【0003】共用サブルーチンについては、文献「マネージング プログラムス アンドライブラリズ イン エーアイエックス バージョン3 フォー リスクシステム/6000プロセッサ: アイビーエム ジャーナル オブ リサーチ アンドデベロップメント, 34巻1号 1990年1月刊 エム. エイ. アウスランダー著」(Managing programs and libraries in AIX Version 3 for RISC System/6000 processors: IBM Journal of Research and Development Vol. 34, no1. January 1990: M. A. Auslander)、文献「シェアード ライブラリ イン サンオーエス: ユーゼニックス プロシーディング 1987年夏, ロバート エイ. ジンゲル等著」(Shared Libraries in SunOS: USENIX proceeding Summer 1987: Robert A. Gingell et al.) に詳しく記載されている。

【0004】この共用サブルーチンを、オンラインで入れ替える必要が生じる場合がある。例えば、システム動作中のサブルーチンのバージョンアップを行う場合や、共用サブルーチンのデバッグを行う場合である。そのため従来のオペレーティングシステムでは、例えば特開平4-39739に示されるような、共用サブルーチンの入れ替え処理を行うコマンド及びシステムを用意している。

【0005】この特開平4-39739においては、サブルーチン交換管理手段は、何れのタスクがどのサブルーチンを判別する呼出判別手段を有しており、該呼出判別手段により走行中のタスクによって現在サブルーチンが使用されているかどうかを判定する。もしくは該呼出判別手段の替わりにどのサブルーチンが現在メモリにマッピングされているかを管理するサブルーチンマッピング情報記録部を持ち、これからサブルーチンの使用状況を調べる。そして共用サブルーチンに含まれる全てのサブルーチンが使用されていない時には入れ替え処理を行

い、タスクにより使用されているサブルーチンがある時は、そのサブルーチンを使用しているタスクが終了するまで待つか、そのタスクを強制終了させてから共用サブルーチンを入れ替える。また、タスクが入れ替え途中のサブルーチンをコールしようとする場合には、タスク管理がそのコールを中止しタスクを強制終了させるか、そのコールを延期する。これにより、システムの全タスクを全停止させることなく、また、タスクの実行結果に不整合が起こさずに共用サブルーチンを入れ替えることができる。

#### 【0006】

【発明が解決しようとする課題】特開平4-39739で述べられている様な従来のサブルーチン入れ替え処理は、サブルーチンを実行しているタスクに対する処理が、タスク強制終了若しくはタスク終了待ちのどちらかに固定されているため、サブルーチン入れ替えがシステムに大きなオーバーヘッドをもたらすという問題があった。

【0007】第一の例としては、異なる性質乃至は処理形態のタスクが混在するシステムにおけるサブルーチン入れ替え処理である。一般にシステムを構成するタスクの性質は処理の形態により2種類に分けることが出来る。1つはユニックスにおけるデーモンプロセスのように、無限ループを繰り返して何らかの処理を継続し、基本的には自らは終了することがないような性質のタスクであり、以下この様なタスクを「デーモン型タスク」と呼ぶ。「デーモン型タスク」は、他のタスクや計算機の入出力装置から処理要求を受け付け、その処理を行うことが多い。また、他のタスクや計算機の入出力装置からの処理要求が存在しない場合は、タスクは実行状態のまま次の要求が到着するまで休止状態となる。もう1つは、処理が終ると自ら終了する性質を有するタスクであり、以下この様なタスクを「通常タスク」と呼ぶ。

【0008】これら「デーモン型タスク」と「通常タスク」の2種類のタスクが混在するシステムにおいては、従来の入れ替え処理では以下のような問題点がある。即ち、共用サブルーチン入れ替え時に、サブルーチン入れ替え処理部の処理がタスクの強制終了に固定されている場合、「通常タスク」がコールするサブルーチンを入れ替える際に、「通常タスク」が以後サブルーチンをコールしなくとも、そのタスクを強制終了する。これは、それまで実行していた「通常タスク」の処理結果を無効とし、始めから処理を際実行しなければならず、処理効率が低下する。

【0009】また逆に、共用サブルーチン入れ替え時に、サブルーチン入れ替え処理部の処理がタスクの処理終了待ちに固定されている場合、「デーモン型タスク」がコールするサブルーチンを入れ替える際に、「デーモン型タスク」は終了することはあり得ないにもかかわらず、そのタスクの終了を待ち続ける。従って、この待ち時間は無駄になる。

【0010】第二の例は、「排他制御命令を有するタスク」と「排他制御命令を有さないタスク」が混在するシステムにおけるサブルーチン入れ替え処理である。「排他制御命令を有するタスク」とは、共有データ、共有データベースなどを排他的にアクセスするために、排他制御命令を使用するタスクである。排他制御命令には、共有データを独占するロックと、共有データを解放するアンロック命令がある。「排他制御命令を有さないタスク」とは、排他制御命令を使用しないタスクである。

10 【0011】これら2種類のタスクが混在するシステムにおいては、従来の入れ替え処理では次のような問題点がある。共用サブルーチン入れ替え時に、サブルーチン入れ替え処理部の処理がタスクの強制終了に固定されている場合、「排他制御命令を有するタスク」がコールするサブルーチンを入れ替える際に、該タスクが共有データをロックしている場合でもこのタスクを強制終了するため、該タスクが占有していた共有データをアンロックするタスクが存在しなくなる。そのため、以後この共有データを使用しようとするタスクはすべて共有データを使用できない状態に陥る。

20 【0012】また逆に、共用サブルーチン入れ替え時に、サブルーチン入れ替え処理部の処理がタスクの処理終了待ちに固定されている場合、「排他制御命令を有さないタスク」がコールするサブルーチンを入れ替える時、「排他制御命令を有さないタスク」はデッドロック状態になることはあり得ないにもかかわらず、そのタスクの終了を待ち続ける。従って、この待ち時間は無駄である。

30 【0013】このように、公知の共用サブルーチン入れ替えシステムでは、捜査の対象となるタスクの性質を考慮することがなかったため、タスクに対する処理とタスク種別の組み合わせによっては、システムにとって大きなオーバーヘッドが発生するという問題点があった。

【0014】本発明の目的は、共用サブルーチンを入れ替える際に、上記の如きオーバーヘッドを削減し効率の良い共用サブルーチン入れ替えシステムを提供することである。

#### 【0015】

40 【課題を解決するための手段】上記目的を達成するため、本発明の共用サブルーチン入れ替えシステムは、ユーザのプログラムを実行するタスクと、タスク間で共有して実行可能なプログラムである共有サブルーチンと、該共用サブルーチンを格納するサブルーチン格納部と、当該サブルーチン格納部に格納されるサブルーチンを入れ替えるサブルーチン入れ替え手段と、タスクの実行開始処理と実行終了処理を行なうタスク管理手段を有する計算機システムにおいて、上記サブルーチン入れ替え手段は、サブルーチンの入れ替えコマンドが実行された時にサブルーチンがタスクにより使用中であった場合に、  
50 当該タスクに対する対処をタスクの性質に基づき指定す

るサブルーチン入れ替え処理指示記録手段を有し、上記タスク管理手段は、タスクがコールしようとするサブルーチンについて入れ替え処理が実行中かどうかを判定し、入れ替え処理中ならばそのタスクの処理を中断、または強制終了し、上記サブルーチン入れ替え手段は、入れ替えようとするサブルーチンの使用状況を判定し、サブルーチンがタスクに使用されておらず、かつサブルーチン入れ替え処理記録手段に指定がない場合、入れ替え処理を行ない、これ以外の場合には、指示に従ってタスクの強制終了を行うか又は、タスクが終了して入れ替え処理可能になるまでサブルーチン入れ替え処理を延期することを特徴としている。

#### 【0016】

【作用】本発明によれば、共用サブルーチン入れ替え処理を行うときには、現在実行中のタスクがその共用サブルーチンを使用していないか確認し、もし使用していれば、タスク毎にその性質に基づき指定されたサブルーチン入れ替え処理指示記録部の指示に従って、タスクの終了を待つか、タスクを強制終了させるかを選択する。

【0017】上記により、例えば「通常タスク」「デーモン型タスク」の如き性質の異なるタスクが混在するシステムにおいてサブルーチン入れ替えが行われるとき、「通常タスク」が実行中の場合は、タスクの終了までサブルーチン入れ替えが延期される。このため、「通常タスク」の途中までの処理結果が、無駄に破棄されることがなくなる。また、「デーモン型タスク」が実行中の場合には、すぐにタスクを強制終了させサブルーチン入れ替えを行う。このため、サブルーチン入れ替え処理のために「デーモン型タスク」の終了を無駄に待つような状況が発生しなくなり、効率の良い共用サブルーチン入れ替えシステムを実現することが出来る。

#### 【0018】

【実施例】本発明における一実施例を、「通常タスク」及び「デーモン型タスク」が混在するシステムにおけるサブルーチン入れ替えシステムの例で説明する。このシステムでは、サブルーチン入れ替え時に「通常タスク」が実行中の場合は、タスクの終了までサブルーチン入れ替えを延期する。また、「デーモン型タスク」が実行中の場合には、すぐにタスクを強制終了させサブルーチンを入れ替える。

【0019】まず図1に、本発明の共用サブルーチン入れ替えシステムの構成を示す。タスク1、2は、ユーザが記述したプログラムやシステムが定めたプログラム処理を行い、必要な場合には共用サブルーチン31~33を呼び出して実行する。各々のタスクは一意に区別可能なようにタスクの識別情報を持っている。タスクの識別情報としては、タスク番号等がある。なお、図1において「破線」矢印はデータの移動を示し、「実線」矢印は制御情報の移動を示している。

【0020】共用サブルーチンは、タスク間で共用され

自由に実行可能なプログラムである。共用サブルーチンには、内部に単一のサブルーチンしか持たないシングルエントリサブルーチンと、複数のサブルーチンを保持するマルチエントリサブルーチンがある。本実施例では、全ての共用サブルーチンがシングルエントリサブルーチンであるとして説明するが、本発明はマルチエントリサブルーチンの場合でも適用できることは明かである。サブルーチン31~33は、タスクの識別情報と同様に一意に区別可能な様にサブルーチンの識別子を持っている。このサブルーチンの識別子としては、サブルーチン番号がある。本実施例においては、共用サブルーチンは「load」コマンドによりロード、「deload」コマンドによりデロードされる。ロードとはディスク等外部記憶装置上にある共用サブルーチンのファイルをメモリ上にコピーすることである。また、デロードとは、ロードされた共用サブルーチンをメモリ上から消すことである。また、タスク開始処理は「queue」、終了処理は「abort」というコマンドで実現されているものとする。

【0021】コマンドインタプリタ6は、ユーザ13がマウス125やキーボード126で入力したコマンドを解釈する。コマンドインタプリタ6は、「load」コマンド、「deload」コマンドを受け取った時には、解釈したコマンドと引数をサブルーチン入れ替え手段15に渡す。また、「queue」コマンド、「abort」コマンドを受けた時は、タスク管理手段14にコマンドと引き数を渡す。なお「load」コマンドでは共用サブルーチンのファイル名とサブルーチン識別子を、「deload」コマンドではサブルーチン識別子を引き数として指定する。

【0022】共用サブルーチンの入れ替えを行うサブルーチン入れ替え手段15は、サブルーチン入れ替え部4、サブルーチン入れ替え処理指示記録部5、サブルーチン使用記録部9、サブルーチン入れ替え記録部11を有している。サブルーチン入れ替え手段15はコマンドインタプリタ11からコマンドとサブルーチン識別子を受けとると、サブルーチン入れ替え部4によって「load」「deload」コマンドを実行する。

【0023】タスクの走行状態を管理するタスク管理手段14は、タスクの開始処理と終了処理を行うタスク管理部10を有しており、コマンドインタプリタ6や、サブルーチン入れ替え部4から呼び出される。「queue」、「abort」コマンドを受け取ったタスク管理手段14では、引数として指定された実行開始するタスクや、実行終了するタスクのタスク番号に基づき、タスク管理部10が指定されたタスク番号を持つタスクの開始処理と終了処理を行う。

【0024】また、サブルーチン格納部3は、ディスク等外部記憶装置に存在するサブルーチンファイル格納部8に格納された共用サブルーチンのファイルをロードするためのメモリ上の領域を表している。

【0025】図3に、上記本発明における一実施例を計

算機12上で実現したときの構成例を示す。タスク1、2と、共用サブルーチン入れ替え部4がディスク123からメモリ122に読み出される。その後メモリ122上にタスク状態記録部7、サブルーチン入れ替え記録部11、サブルーチン入れ替え処理指示記録部7、サブルーチン格納部3のための領域を確保する。そして、これらのタスクやサブルーチン入れ替え部をプロセッサ121が実行する。図3では、全てのサブルーチン31~33やタスク1、2及び共用サブルーチン入れ替え部4が連続したメモリ上に存在する構成として表されているが、これらは不連続のメモリ上に存在することも可能であるし、また全てが同時にメモリ上に存在する必要はなく必要に応じてメモリ上に存在すればよい。更に、本実施例の計算機は入出力装置としてディスプレイ124、マウス125、キーボード126、プリンタ127を備えているが、入出力装置はこれらに限られないし、又これらを全て備えている必要はない。

【0026】図4に、サブルーチン入れ替え記録部11の構成について示す。サブルーチン入れ替え記録部11は、各サブルーチン31~33に対応してサブルーチンの実行が許可されているか否かを示す領域111~113を有している。各領域111~113は対応するサブルーチンが使用許可の場合には「1」、使用不許可の場合には「0」が記録される。従って、共用サブルーチンがロードされていれば「1」、デロードされていれば、「0」がそれぞれ格納される。

【0027】サブルーチン使用記録部9は図5に示すように、各タスク1、2がリンクする全共用サブルーチンを記録する領域の列92~94と、共用サブルーチンに対応してサブルーチン識別子を記録する列91を有している。本実施例ではサブルーチン使用記録部9はビット列で実現し、タスクがリンクするサブルーチンを記録する列の各々の値は、サブルーチンが使用状態である場合には「1」、未使用状態である場合には「0」が記録される。サブルーチン使用記録部9の内容は全て前もって初期化されているものとする。例えば、タスクのコンパイルあるいは共用サブルーチンのリンク時に、タスクが使用する共用サブルーチンのサブルーチン識別子とタスク番号をユーザに指定させ、これをサブルーチン使用記録部のファイルに記録しておくことによっても初期化が可能である。

【0028】図6に、サブルーチン入れ替え処理指示記録部5の構造について示す。サブルーチン入れ替え処理指示記録部5は、各タスク1、2が走行状態の時に、タスクがリンクするサブルーチンが「deload」される場合、そのタスクを強制終了させるか、そのサブルーチンの終了まで「deload」を停止するかの指示をタスク毎に記録する。前者の場合は「KILL」、後者の場合は「WAIT」が各タスクの対応する領域に記録される。

【0029】図7に、タスク状態記録部7の構造について示す。タスク状態記録部7は、各タスク1、2毎に該タ

スクが走行状態か、停止状態かを示す領域71~73を有している。そして各領域には対応するタスクが走行状態ならば「1」、停止状態ならば「0」が記録される。

【0030】以下、タスクが実行を開始した後、共用サブルーチンを実行し、タスクの処理を停止するまでの処理手順を説明する。その後、サブルーチンのロード、デロードを行うコマンド処理の処理手順について説明する。

#### 【0031】A. タスクの処理手順

以下図1において、ユーザのコマンド「queue」、または、システムの他の管理部から「queue」がコールされてタスク1が実行を開始し、続いてタスクが終了する手順について説明する。なおここでは図4~7に示すように、タスク1はサブルーチン31、32をリンクするものとして説明する。

【0032】タスク1を正常に動作させるためには、タスク1の使用するサブルーチン31、32が、事前にサブルーチン格納部3にロードされなければならない。ここでは、システムを操作するユーザがコマンドインタプリタ6にサブルーチン31、32の「load」を発行することによって、既にサブルーチン31、32がロードされているものとする。

【0033】タスク実行開始直後、最初に処理の制御はタスク管理部10に移り、このタスク管理部10によりコマンド「queue」が実行される。この「queue」コマンドのフローチャートを図8に示す。

【0034】(1) タスクの種別を判定し、サブルーチン入れ替え処理指示記録手段5の該当タスクの領域に、タスクがコールするサブルーチンが「deload」される場合の処理を記録する。「デーモン型タスク」の場合は、タスクを強制終了させないと入れ替えられないので、「KILL」を記録する。「通常タスク」の場合はその終了まで「deload」を停止するだけでよいので「WAIT」を記録する(801)。

【0035】(2) サブルーチン使用記録部9から、当該タスクの使用するサブルーチン番号を1つ読み込む(802)。

【0036】(3) サブルーチン入れ替え記録部11により、当該サブルーチン番号に対応する領域が1(使用許可)か否かを調べる(803)。0(使用不許可)ならば、タスクは実行を継続することは不可能なので、エラー処理を行い終了する。1(使用許可)ならば、処理(4)を実行する。

【0037】(4) 上記(1)、(2)、(3)の処理を、サブルーチン使用記録部9に記録されているサブルーチン識別子全てについて繰り返して行った後(804)、タスク状態記録部の該当タスクの領域を1(走行状態)にセットする。

【0038】(5) タスクの実体を生成する(805)。

【0039】ここで、処理(1)におけるタスクの種別

の判定方式の一例を示す。「デーモン型タスク」は、他のタスクからの処理要求を読み込むための命令を含んでいる。ユニックスではシグナル、ソケット、共用メモリなどの機能を使用して処理要求を読み込む。そして処理要求が存在しない場合は「デーモン型タスク」は休止状態となる。本実施例では、要求読み込みと、要求がない場合に休止状態となる動作を「qwait」というシステムコールにより行うものとする。通常「デーモン型タスク」では、この「qwait」が無限ループで囲まれている。そこで、タスクのソースプログラムをコンパイルする際に、ループで囲まれた「qwait」を検出した場合、そのタスクは「デーモン型タスク」と判定することにする。そして、その「デーモン型タスク」と判定されたタスクのオブジェクトファイルのヘッダに、タスク種別を表わす領域を生成し「デーモン型タスク」と記録する。

【0040】一方、ループで囲まれた「qwait」が検出できなかった場合、そのタスクは「通常タスク」と判定し、オブジェクトコードのヘッダに「通常タスク」と記録する。そして(1)では、オブジェクトファイルのヘッダにあるタスク種別を表す領域を参照することにより、タスクが「デーモン型タスク」であるか「通常タスク」であるかを判別する。

【0041】タスク管理部10の処理が終了すると、処理は生成されたタスクに移る。そしてタスクの処理が終了すると、タスク自身が「abort」をコールしてタスク管理部の終了処理が行われる。「abort」コマンドのフローチャートを図9に示す。

【0042】(1)タスク状態記憶部の当該タスクの領域に、停止状態と記録する(901)。

【0043】(2)タスクの実体を消滅する(902)。

【0044】本実施例では、共用サブルーチンの最初のメモリへのロードはコマンドインタプリタ6からコマンドが入力されることで行うものとしている。しかし、計算機の立ちあげ時に、全てのタスクについてサブルーチン使用記録部9に登録されたサブルーチンをロードするコマンドを発行させる方式も考えられる。例えばユニックスではrcコマンドを用いて、計算機の立ちあげ時にロードコマンドを実行する方式がある。rcコマンドについては「ユニックス4.3ビーエスディー(BSD)の設計と実装(S.J. Leffler等著、丸善株式会社刊)」に詳しい記載がある。

【0045】

B. ロード、デロードのコマンドの処理手順

図1において、共用サブルーチンのロード、またはデロードコマンドが入力された場合を例として本実施例を説明する。

【0046】まず、ユーザがキーボード125やその他の入力装置にコマンド「load」、「deload」を入力すると、コマンドインタプリタ6は、コマンドとその引き数をサブルーチン入れ替え手段15に送り、制御をサブルー

チン入れ替え手段15に移す。サブルーチン入れ替え手段15では該送られたコマンドをサブルーチン入れ替え部4で処理する。サブルーチン入れ替え部4の「deload」の動作を表すフローチャートを図10に示す。

【0047】(1)まず、サブルーチン入れ替え記録部11の当該サブルーチン番号の領域を0(使用不可能)にセットする(1001)。

【0048】(2)サブルーチン使用記録部9の当該サブルーチン番号の行から、あるタスクについて、そのタスクは当該サブルーチンを実行するか否かを検索する(1002)。実行する場合は(3)に移る。そうでなければ(5)に移る。

【0049】(3)そのタスクが実行状態であるかどうか、タスク状態記録部7を検索する(1003)。実行状態であれば、(4)に移る。そうでなければ(5)に移る。

【0050】(4)実行中のタスクに対する処理をどうするかを、サブルーチン入れ替え指示処理記録部5の該当タスクの領域から調べる(1004)。該当タスクの領域が「KILL」であればタスク管理手段14によりタスクを強制終了させ、「WAIT」であれば、該当タスクが終了するまで待つ。その後(5)に移る。

【0051】(5)上記(2)、(3)、(4)の処理をタスク番号全てについて繰り返す(1005)。

【0052】(6)該当サブルーチンに割り当てられたメモリ領域を解放する(1006)。

【0053】このようにして古い共用サブルーチンをメモリから削除する。

【0054】次に、図11にサブルーチン入れ替え部の「load」動作のフローチャートを示す。

【0055】(1)サブルーチン格納部3のうち、サブルーチン識別子により指定された未使用のメモリ領域を当該サブルーチンに割り当てる(1101)。その後、サブルーチンファイル格納部8から、引き数で指定されたサブルーチンファイルを読み込み、割り当てられたメモリ領域にロードする(1102)。

【0056】(2)サブルーチン入れ替え記録部7のサブルーチン番号の領域を「1」(使用可能)にセットする(1103)。

【0057】この様にして、新たな共用サブルーチンに入れ替える。

【0058】上述の本発明における一実施例においては、「load」、「deload」コマンドが実行不可能な場合にも、そのまま共用サブルーチン入れ替え処理は継続できるようにになっていた。しかし、共用サブルーチンのロード、デロード処理よりもタスクの実行を妨げないことの方が重要であるシステムの場合、「queue」コマンド実行時にこのような処理を優先させるタスクに対しては、サブルーチン入れ替え処理指示記録部5の対応する領域に「ABORT」を記録する。そして、「deload」コマ



ンド実行の際、サブルーチン入れ替え処理指示記録部5に「ABORT」が書かれているタスクの場合、「deload」コマンドの処理を強制終了させることにする。これにより、処理を優先させなければならないタスクが存在するシステムにおいても、信頼性の高い共用サブルーチン入れ替え処理を行うことが出来る。

【0059】また、前記本発明の一実施例において、タスク毎のサブルーチン入れ替え処理時の動作指示は、オブジェクトファイルを元に自動的に行うものとしていた。これに対して、サブルーチン入れ替え処理時のタスクに対する処理の指示を、共用サブルーチン入れ替え処理が発生したときに、必要に応じてユーザから与えるようにすることも可能である。この実施例の構成を図12に示す。

【0060】この共用サブルーチン入れ替えシステムは、サブルーチン入れ替え処理記録部5の代わりにメッセージ表示部16を持つ。そして、「deload」コマンドの実行中に入れ替え対象のサブルーチンを使用しているタスクを発見すると、メッセージ表示部16により「タスクを強制終了させる」か「タスクが終了するのを待つ」か「コマンドを終了する」といった選択肢を画面に表示してユーザに処理を選択させる。そしてユーザの選択に従って「deload」コマンドの処理を続行する。上記動作は、図10のステップ1004の処理において、メッセージ表示部が当該タスクに対する処理の選択肢をユーザに示し、ユーザの選択に従って当該タスクを強制終了するか、乃至は当該タスクの終了を待つか、あるいはコマンドを終了するかを選択するように変更することで実現できる。

【0061】これにより、不用意なタスクの終了乃至はサブルーチン入れ替え処理の遅延を防ぎ、効率の良い共用サブルーチン入れ替え処理を行うことが出来る。

【0062】更に、前記本発明の一実施例においては、共用サブルーチンはシングuentリサブルーチンであるとして説明してきた。しかし、ユニックスなどの共用ライブラリは、一般にマルチuentリサブルーチンであり、この場合について以下記述する。

【0063】共用ライブラリの場合には、サブルーチン毎ではなくライブラリ毎にサブルーチン識別子としてサブルーチン番号を割り当てる。そして、サブルーチン使用記録部9は各タスクがどのサブルーチンを使用するかをサブルーチン毎ではなくライブラリ単位で記録する。また、「load」「deload」コマンドの入れ替え指定単位は、各サブルーチン単位ではなくライブラリ単位とする。このようにすることにより、マルチuentリの共用サブルーチンライブラリをオンラインで効率良く入れ替えるシステムを実現することができる。

## 【0064】

【発明の効果】本発明によれば、共用サブルーチン入れ替え処理を行うときには、現在実行中のタスクがその共用サブルーチンを使用していないか確認し、もし使用していれば、タスク毎に指定されたサブルーチン入れ替え処理指示記録部の指示に従って、タスクの終了を待つか、タスクを強制終了させるかを選択する。

【0065】上記により、例えば「通常タスク」「デーモン型タスク」が混在するシステムにおいてサブルーチン入れ替えが行われるとき、「通常タスク」が実行中の場合は、タスクの終了までサブルーチン入れ替えが延期される。このため、「通常タスク」の途中までの処理結果が、無駄に破棄されることがなくなる。また、「デーモン型タスク」が実行中の場合には、すぐにタスクを強制終了させサブルーチン入れ替えを行う。このため、サブルーチン入れ替え処理のために、「デーモン型タスク」の終了を無駄に待つような状況が発生しなくなる。

## 【図面の簡単な説明】

【図1】本発明のサブルーチン入れ替えシステムの構成図である。

【図2】本発明の別のサブルーチン入れ替えシステムの構成図である。

【図3】本発明のサブルーチン入れ替えシステムを計算機上に実現した構成図の一例である。

【図4】サブルーチン入れ替え記録部の構成図である。

【図5】サブルーチン使用記憶部の構成図である。

【図6】サブルーチン入れ替え処理指示記憶部の構成図である。

【図7】タスク状態記憶部の構成図である。

【図8】本発明にかかる実施例の動作を示すフローチャートである。

【図9】本発明にかかる実施例の動作を示すフローチャートである。

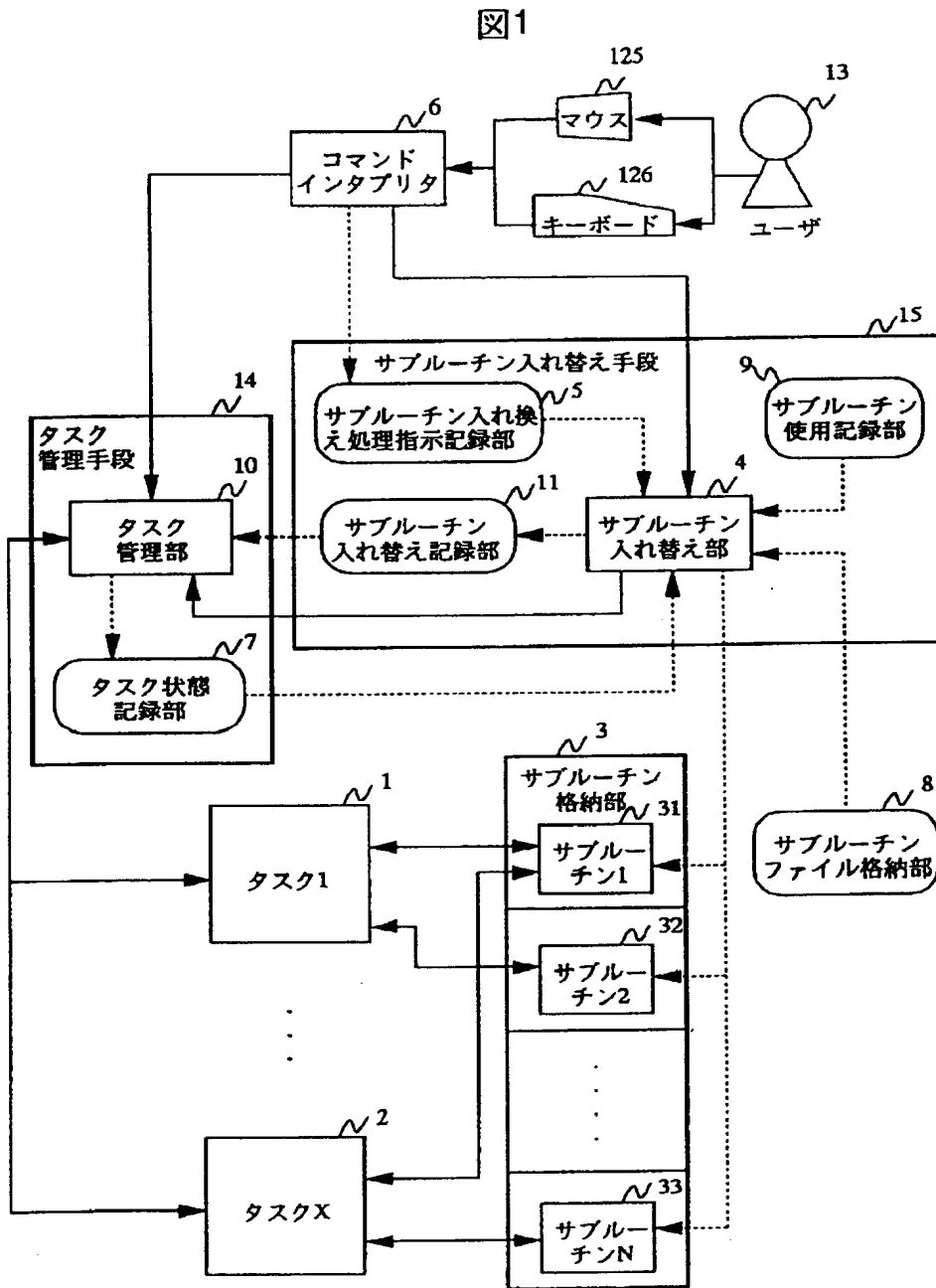
【図10】本発明にかかる実施例の動作を示すフローチャートである。

【図11】本発明にかかる実施例の動作を示すフローチャートである。

## 【符号の説明】

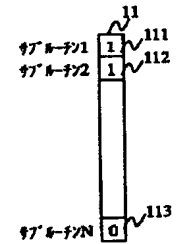
1、2…タスク、3…サブルーチン格納部、4…サブルーチン入れ替え部、5…サブルーチン入れ替え処理指示記録部、6…コマンドインタプリタ、7…タスク状態記録部、8…サブルーチンファイル格納部、9…サブルーチン使用記録部、10…タスク管理部、11…サブルーチン入れ替え記録部、12…計算機、14…タスク管理手段、15サブルーチン入れ替え手段、16…メッセージ表示部、31～33…サブルーチン、125…マウス、126…キーボード。

【図1】



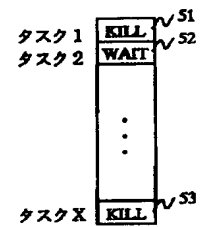
【図4】

図4



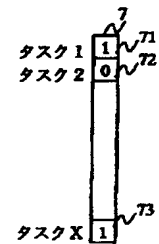
【図6】

図6



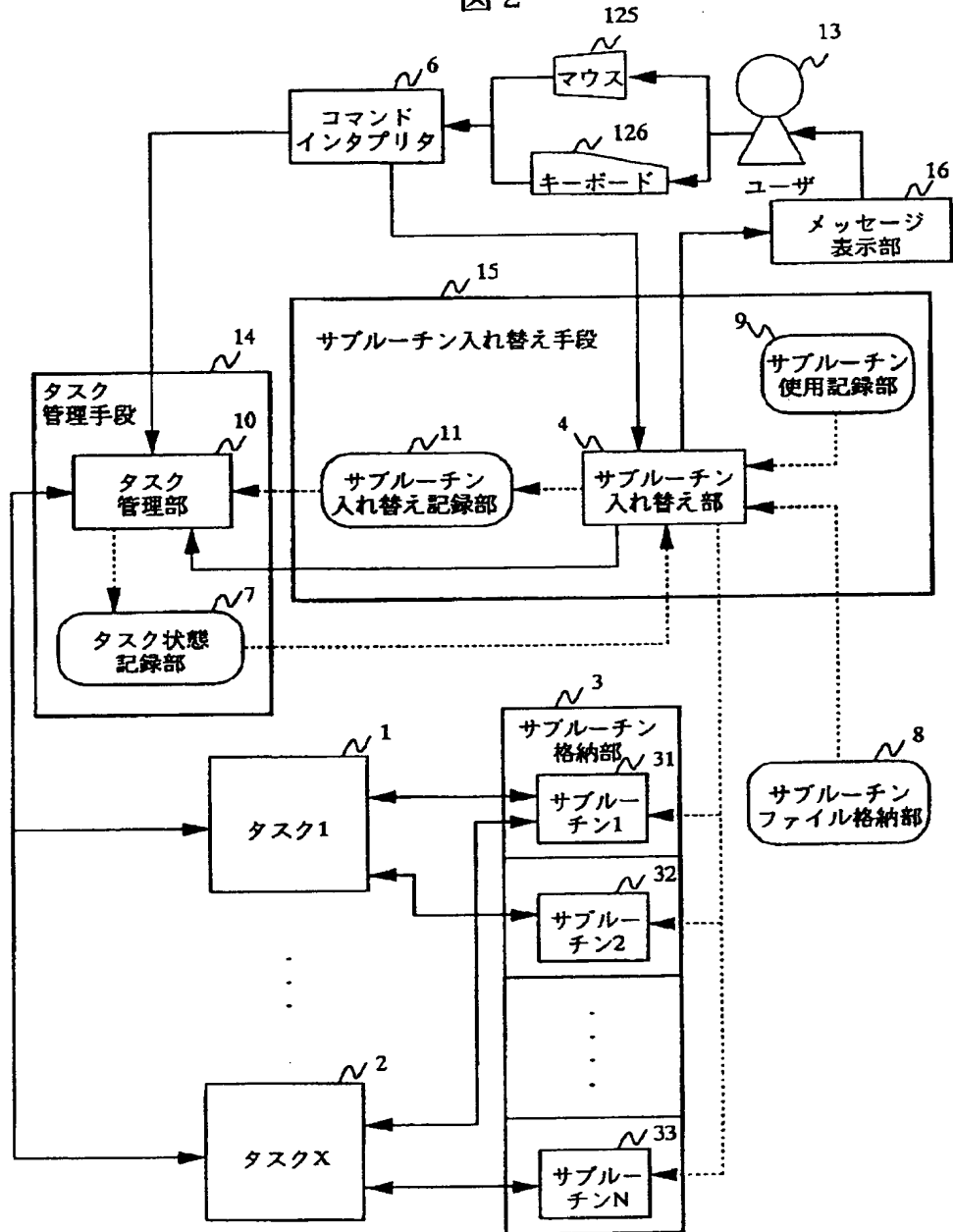
【図7】

図7



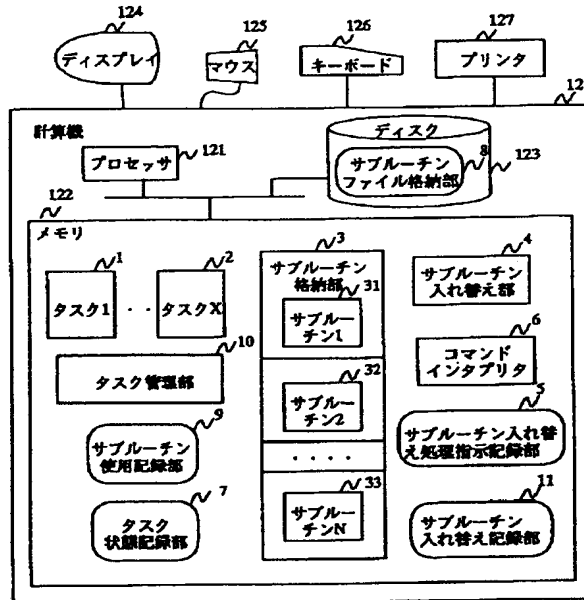
【図2】

図2



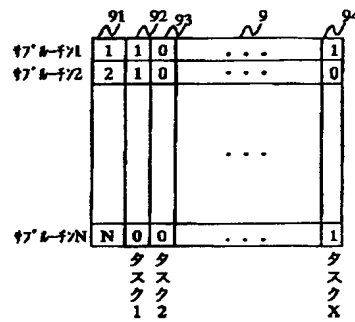
【図3】

図3



【図5】

図5

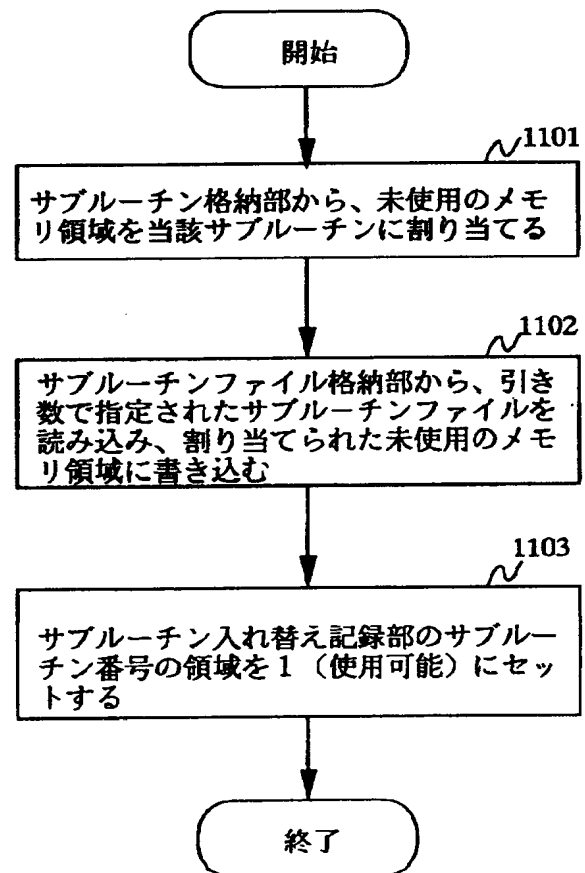
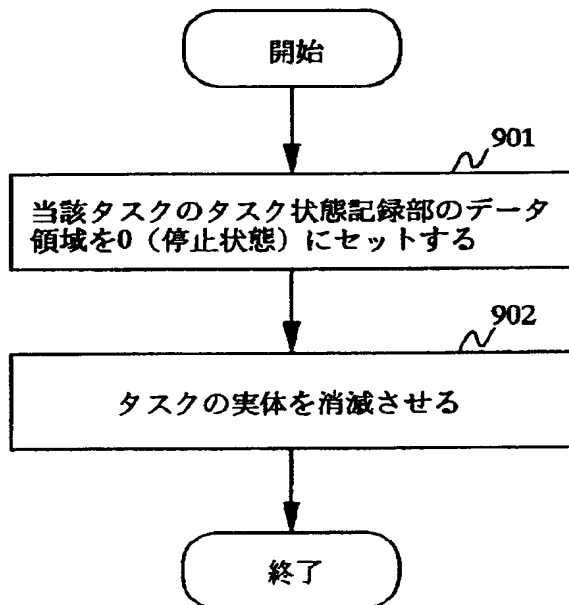


【図11】

図11

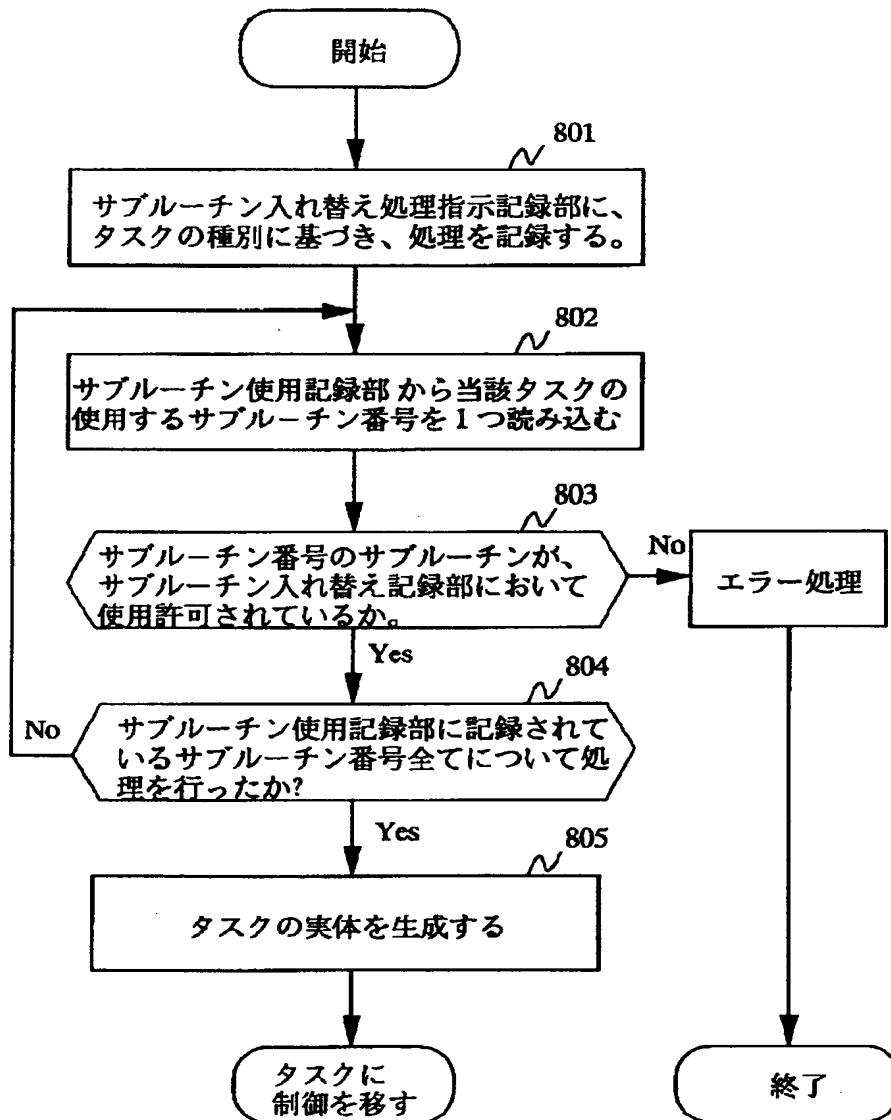
【図9】

図9



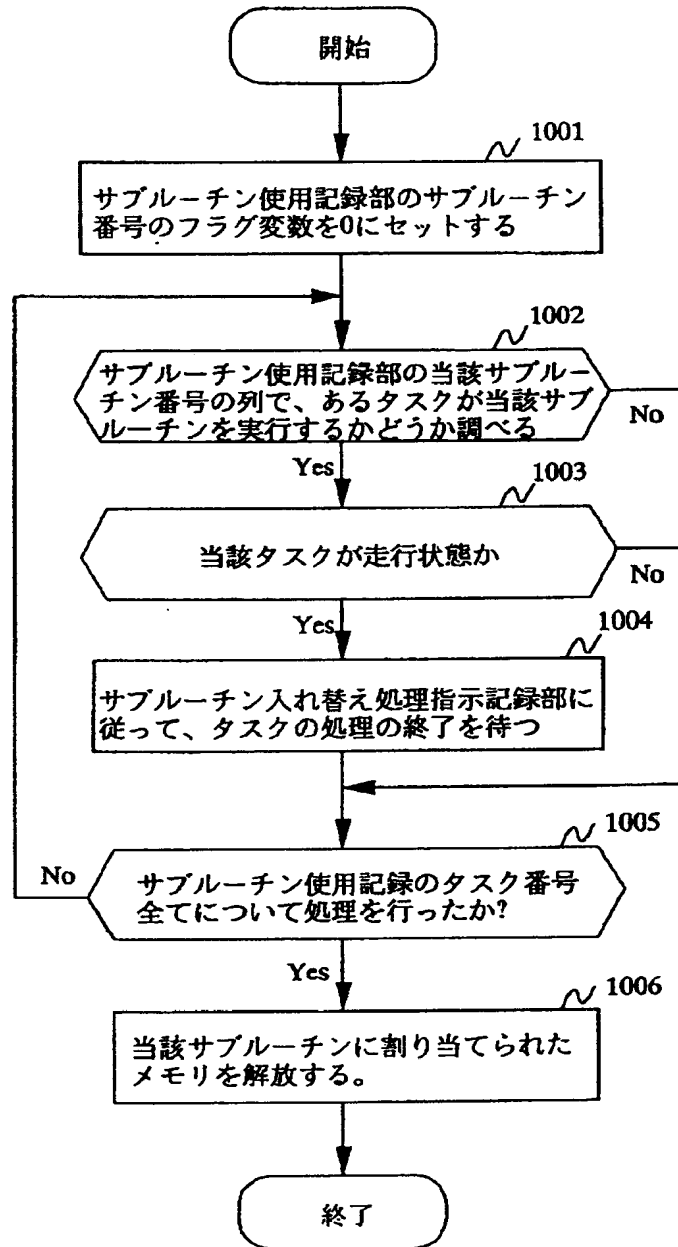
【図8】

図8



【図10】

図10



フロントページの続き

(72)発明者 齊藤 雅彦  
茨城県日立市大みか町七丁目1番1号 株  
式会社日立製作所日立研究所内

(72)発明者 中村 智明  
茨城県日立市大みか町五丁目2番1号 株  
式会社日立製作所大みか工場内